EXHIBIT IV.2 - CALPERS TARGET ARCHITECTURE PRINCIPLES

The following list of enterprise architecture principles provide an overview of the most critical architectural principles that guide the long-term evolution of the CalPERS target architectural environment. QBPs must clearly illustrate how each of these key architectural principles are addressed.

1. Open Standards, Open Technology, and Common Interfaces

Open standards and technology and the use of common interfaces are required.

- Use of open standards such as J2EE, XML, and others allow for ease of integration and provides for a reduction in complexity;
- Use of open technologies reduces reliance on vendor products and improves interoperability;
- Use of common interfaces allows the organization to change vendors and products with out greatly impacting the enterprise;
- Systems that can be quickly reconfigured and extended when conditions change are less expensive to maintain; and
- An open API standard or API abstraction for proprietary component interfaces must be utilized to minimize technology “lock-in”.

2. Multi-Layer Architecture

The enterprise architecture will be multi-layered such that domains are defined and in which the relationship between domains are defined. Components will be defined within each domain with standard inputs, processes, and options. Infrastructure, data, and applications will be designed in such a way that they can be maintained independently of each other.

- QBP must implement an application architecture consistent with the CalPERS target conceptual application architecture;
- Component independence must be defined through interfaces; and
- A component should represent a function or process that cannot be further divided into a reusable function or process.

3. Reduce Integration Complexity

A goal of the architecture is to reduce integration complexity to the extent possible.

- Reduced integration complexity increases the capacity of infrastructure to ‘adapt’; and,
- Lowers the cost of support and the total cost of ownership.

4. Application Partitioning

The logical design of application systems should be highly partitioned into discrete service layers.

- This supports the reuse of components; and,
- It also supports scaling by allowing physical partitioning of functions across servers.

5. Database Partitioning

Databases should have a high degree of logical partitioning.

- This supports the scaling of databases across physical servers and high performance through parallel access across servers.

6. Firm Logical Boundaries

Logical boundaries must be established between the partitions, application, or database, and the logical boundaries must not be violated.

- This supports flexibility and scaling (up and down) of applications and data across or within servers;
- The architecture boundaries should be defined in such a way that the system is broken up into reusable components. Components help simplify complexity, providing both modularity and independence in applications, infrastructure, software, and physical resources; and,
- Architecture should isolate via layers the presentation logic from the business logic, information management logic, security management, and other resources.

7. Message-Based Interfaces

The interfaces between separate application systems must be message-based.

- This supports the integration of heterogeneous servers and system platforms;
- It also supports the concept of autonomous asynchronous execution;
- These interfaces must extend across the value chain to include both customers and suppliers;
- Use synchronous messaging for transactional operations, and asynchronous messaging for non-transactional or long-lived operations;
- Use coarse-grained remote interfaces for synchronous messaging, and,
- Use publish and subscribe to send notifications or information when there are multiple interested parties and/or the publisher may not know the interested parties.

8. Event-Driven Systems

Application systems must be (business) event-driven.

- This allows rapid response to business events and time sequencing of event actions.

9. Highly Granular, Loosely Coupled

Application systems must be 'highly granular' and 'loosely coupled'.

- Highly granular, loosely coupled components are requirements for partitioned, reusable application components.

10. Server Partitioning - "Workloads"

Applications and databases should be physically partitioned on separate servers, in the same location, based on workload.

- Applications with differing characteristics can adversely affect each other's performance.

11. Client/Server and Web Browser Model

Application systems must be implemented using either client/server or web browser for the presentation view.

- This separates user presentation services from applications and information storage. It is a flexible distributed computing model; and,
- It supports flexible physical partitioning of applications and data, scalability, and sharing of information.

12. Appropriate Geographic Partitioning

Applications and databases should be geographically partitioned as appropriate.

- This reduces network bandwidth requirements and results in more responsive applications.

13. Leverage Data Warehousing

Data warehouses shall be leveraged to accelerate decision-making and reduce the development burden.

- Emphasis on multiple data warehouses and using replicated data.

14. Separate OLTP from Data Warehouse

Separate Online Transaction Processing (OLTP) from Data Warehouse (DW) and other end user computing.

- Growth in OLTP is incremental and requirements are predictable. Growth in DW and end user computing has been non-linear and requirements are very difficult to predict.

15. Device Independence

All applications must be capable of supporting different interface types.

- Support true device independence and shared context. As networks and services evolve, it becomes impossible to know who or what is on the other end.

With channels converging and devices multiplying, it's not enough to write multiple interfaces to functionality.

16. Technology Independence

Architecture will be designed in such a way to shield the application from the underlying technology.

- Architecture should be layered in such a way that the technologies that make up the solution can be abstracted or hidden from the application logic;
- Architecture must include standards which support application portability;
- Application programming interfaces (API's) to enable legacy applications to interoperate must be available;
- Middleware must decouple applications from specific software solutions;
- Connectivity between non-public components within an application is not restricted to any technology; and,
- Connectivity between public components with an application is restricted to use technology presently included in CalPERS "as-is" technical architecture.

17. Optimization

- Architecture will be designed and implemented to optimize performance without compromising maintainability, consistency, integrity, and accuracy; and,
- Architecture should be layered without compromising maintainability, consistency, and accuracy.

18. Off-the-Shelf Solutions

Commercial Off-the-Shelf Solutions (COTS) will be utilized to reduce costs and speed application development, delivery, and implementation.

- When additional software components are required, the first option, where possible, will be to first reuse existing technology components; secondly, implement package software; and finally, custom build the solution.

19. Compliance with Security Policies, Rules, and Standards

Protection of CalPERS business and IT assets is a paramount consideration in all IT initiatives.

- Security default is "denial of access";
- Security will be designed into CalPERS infrastructure architecture rather than simply being "added on"; and,
- CalPERS data and related assets will be protected from unauthorized access by implementing authentication and authorization services.

20. Business Continuity

The protection of CalPERS ability to conduct business will be paramount in all IT related activities.

- CalPERS business continuity and disaster planning policies, strategies, practices, and standards will be strictly adhered to in all IT-related initiatives;
- Mission critical IT assets will be given priority in the planning and development of CalPERS business continuity initiatives; and,
- Recoverability, redundancy, and maintainability are key mandatory components of all IT-related initiatives.